Teams Leave App

(In Terminal using Ruby)

Works in Progress

Christopher Chong - March 2022

2021. *Annual leave in Hong Kong – A guide to Employee and Employer.* [image] Available at: <https://zegal.com/blog/post/annual-leave-hong-kong/> [Accessed 30 March 2022].

Objective of the Team Leave App

- The concept behind the app is a field based work problem (from my previous work experience) of conflicting leave date bookings between staff members in the same team.
- There should be at least one staff member per team present in the office at all times (except public holidays).
- For proof of concept, this app design is based on a single team of three or more staff. Also assuming all requested leave dates have not already been taken.
- Allocation of leave entitlements are different for Managers and Team Members.

Main Features of Team Leave App

1. **Request leave (for holidays):** Staff login used for validating user and a date validation before leave is requested. Date validation includes, database checking for double booking of same staff member and for minimum required staff at work.

2. **Delete existing leave (in case of scheduling changes):** Staff login used for validating user, only existing requested leave can be deleted through database checking.

3. **Create new staff member (when new employee joins the team):** New staff member will enter name, role and password and the app will auto-generate a unique Staff ID to be used for requesting and deleting leave as well as update staff name and password. These details will be automatically added to the existing staff database.

Main Features of Team Leave App (cont.)

4. **Update existing staff member name (i.e. Marriage change name) / password (good security practices):** Staff login used for validating user before allowing changes to name or password in the existing database. User input for user name and password would go through valid validation conditions (i.e. password length or empty entries).

5. **Delete existing staff member (in case of Team reshuffling or Team Member leaving):** Staff login used for validating user, only existing staff can be deleted in the existing database.

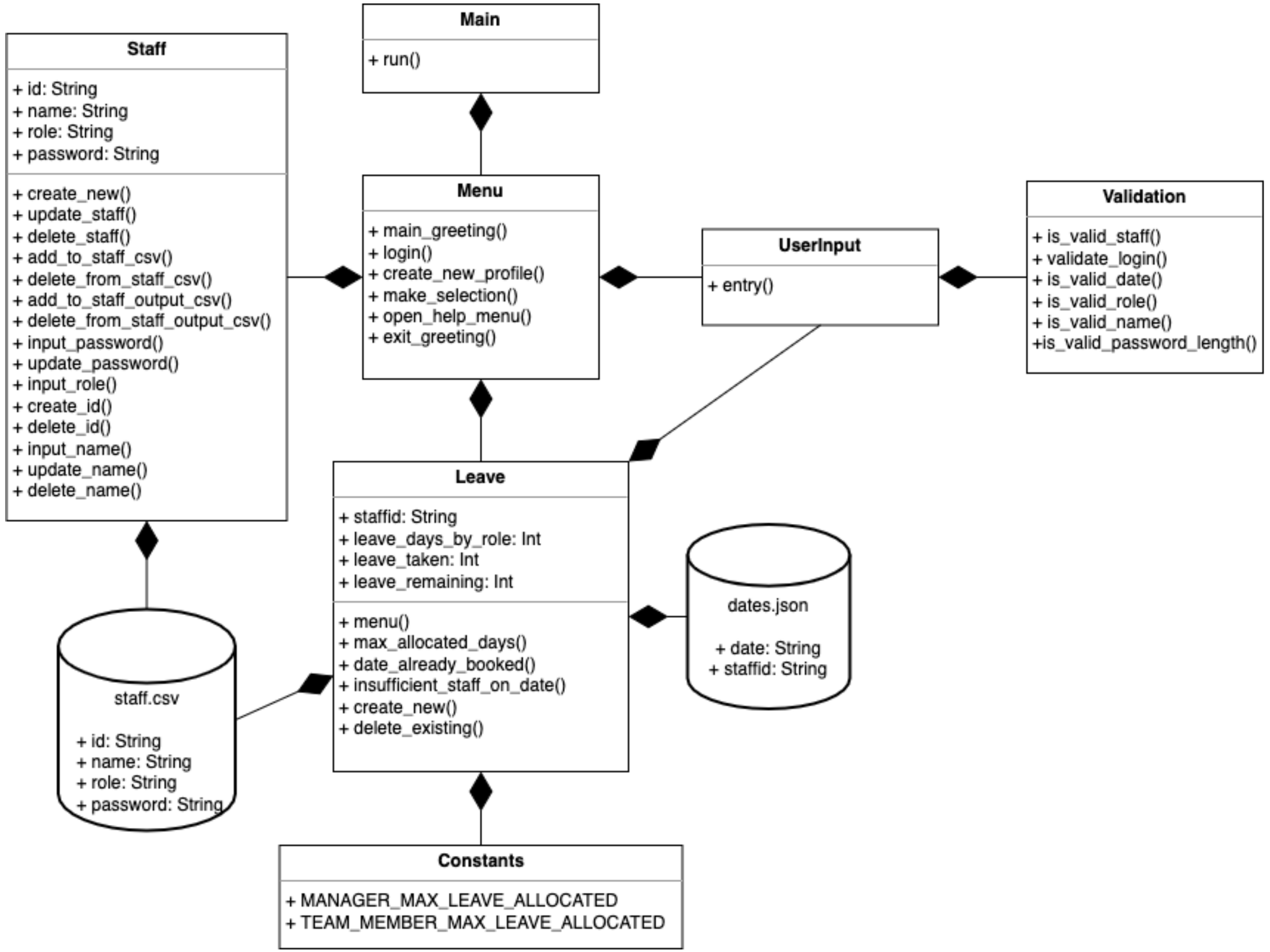
6. **Display remaining leave credit and current leave requested (to allow better leave planning):** Staff login used for validating user and Staff ID used for retrieving leave details from the existing database.

7. **Staff login with password and Staff ID (for security purposes):** Staff ID and password are validated through database checking and confirming they exist and their credentials match.

Quick Demo

Class Diagram (App Structure)

- Everything feeds into Menu (navigation), then feeds into Main.
- Two databases, one for staff details and one for leave dates.
- Each page contains Help menu and Exit option.
- Constants and Validation classes separated from main source code, to allow values to be added/updated from one point of entry.
- User input is in a seperate class to keep the code DRY.



Important parts of the code

Crucial application logic

- Staff ID is used as a unique identifier for staff login and requesting leave to avoid complications arising from staff with identical names (i.e. John Smith)
- Login validation to match Staff ID and password from staff database to allow permissions from verified staff, preventing mishandling of data entry.
- Running RSpec to unit test for creating an instance of staff or leave class, which is then used to populate the staff and dates databases. This is important to ensure correct data formatting throughout the app.

Important parts of the code (cont.)

Crucial application logic

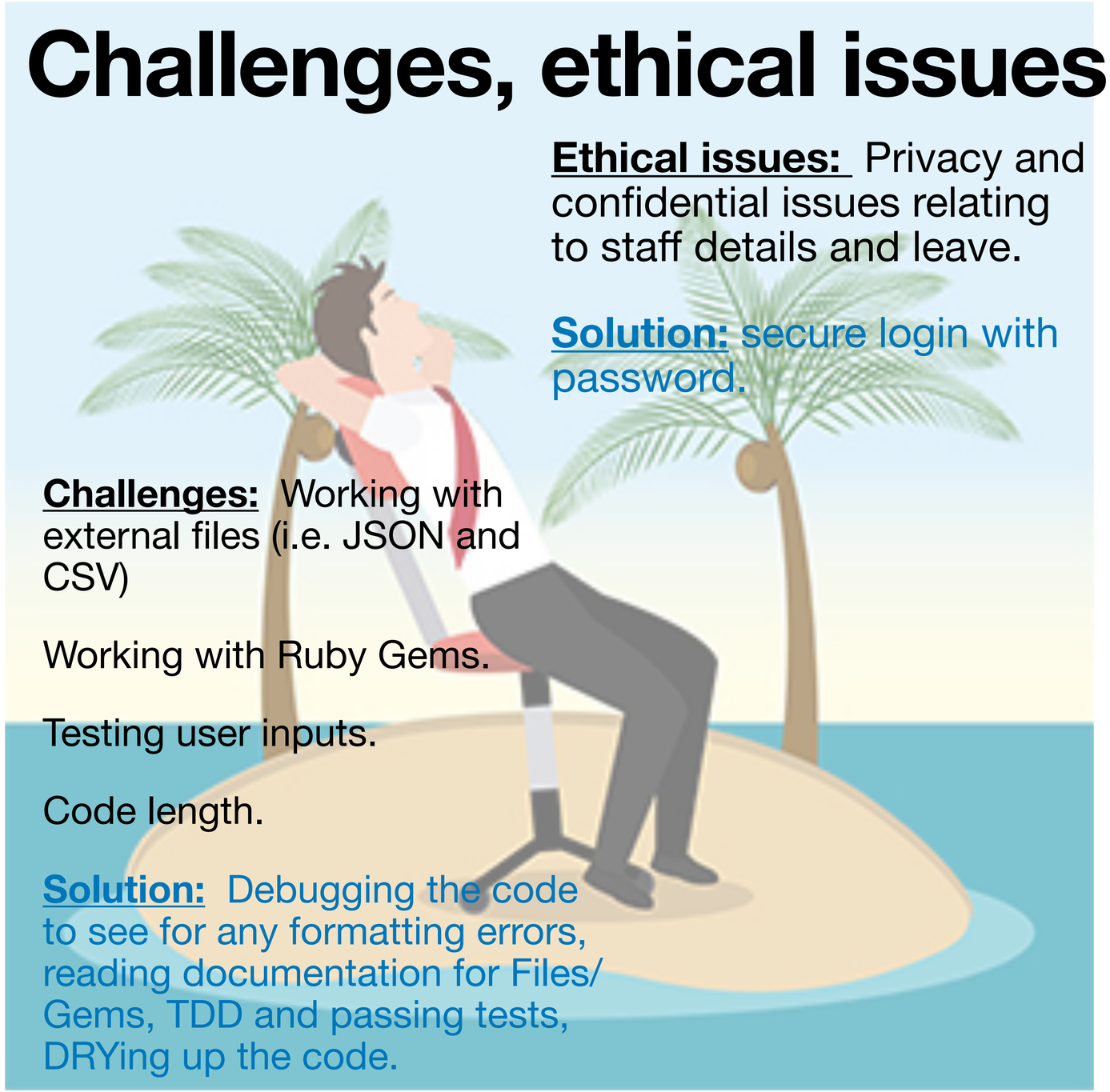
- Validation class is important to have to consolidate all the different validation methods used throughout the app. Allowing any changes or updates in validation conditions to updated in this one class.
- The constants class is created for ease of modifying any constant values used throughout the app in one place. This allows DRYer code to be written, as it prevents duplication of values.
- Reading/Writing/Updating/Deleting to and from staff and dates databases is important because it is used for validation process and single source of information forming the foundation of the Teams Leave app.

Important parts of the code

Ruby Gems

- **rainbow - to highlight important text (e.g. help menu and exit):**
  https://rubygems.org/gems/rainbow
- **table_print - to organise leave dates (when booking leave):**
  https://rubygems.org/gems/table_print
- **terminal-basic-menu - for main menu (to navigate):**
  https://rubygems.org/gems/terminal-basic-menu
- **tty-box - for drawing error boxes (when wrong password entered and confirming changes):**
  https://github.com/piotrmurach/tty-box

Challenges, ethical issues, favourite parts…



Challenges: Working with external files (i.e. JSON and CSV)

Working with Ruby Gems.

Testing user inputs.

Code length.

Solution: Debugging the code to see for any formatting errors, reading documentation for Files/ Gems, TDD and passing tests, DRYing up the code.

Ethical issues: Privacy and confidential issues relating to staff details and leave.

Solution: secure login with password.

Favourite parts: Running the app.

TDD test passing and methods working as expected without errors.

Manually testing the app.

Formatting terminal display.

The app solves a real world issue.

2016. *How to Calculate the Annual Leave Periods Of Part-Time Workers*. [image] Available at: <https://turkishlaborlaw.com/news/legal-news/how-to-calculate-the-annual-leave-periods-of-part-time-workers/> [Accessed 30 March 2022].

Thank you